*Please note that we haven't explained full-fledged annotation based Hibernate project here. Rather we have focused on explaining how bidirectional one to many association is mapped in Hibernate.*

*To know more about annotation based basic Hibernate project please thoroughly go through project* ***' Hibernate5StandaloneWithFullJavaConfig'*** *if you are not familiar with it.*

**What this project dose ?**

This project explains how bidirectional one to many association is mapped in Hibernate by using annotations.

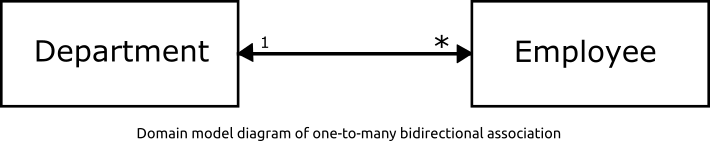
**A short introduction**

In one-to-many association, source entity has a field that stores one or more target entities. The @OneToMany JPA annotation is used to link the source entity with the target entities.

**Bidirectional**→ In this type of association, each entity (i.e. source and target) has a relationship field that refers to each other. We can navigate this type of association from both sides.

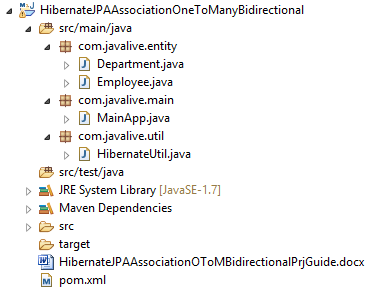
**Bidirectional one-to-many association example**

Following are the domain model and relational model diagrams of one-to-many bidirectional association.

****



**Project Structure**

****

**Jar dependencies**

Add the following jar dependencies for Hibernate and MySQL driver in pom.xml file.

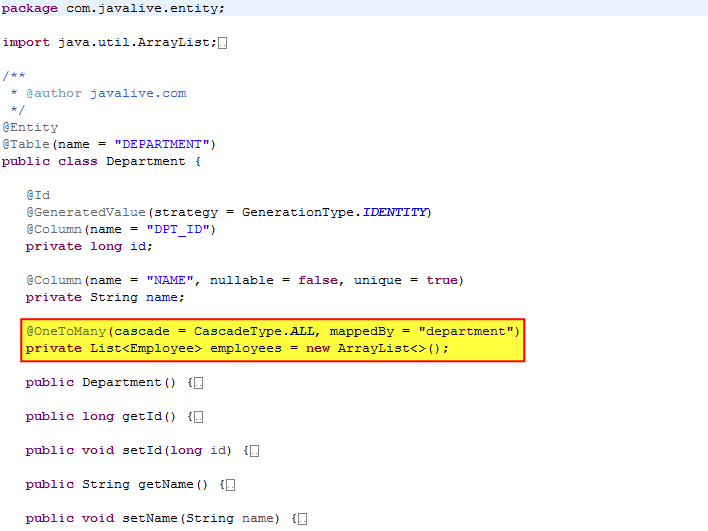
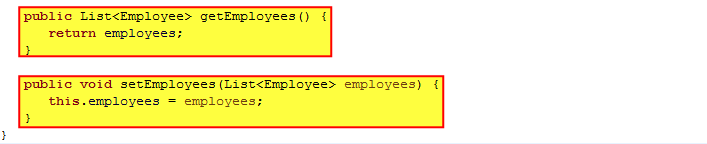


**Entity class**

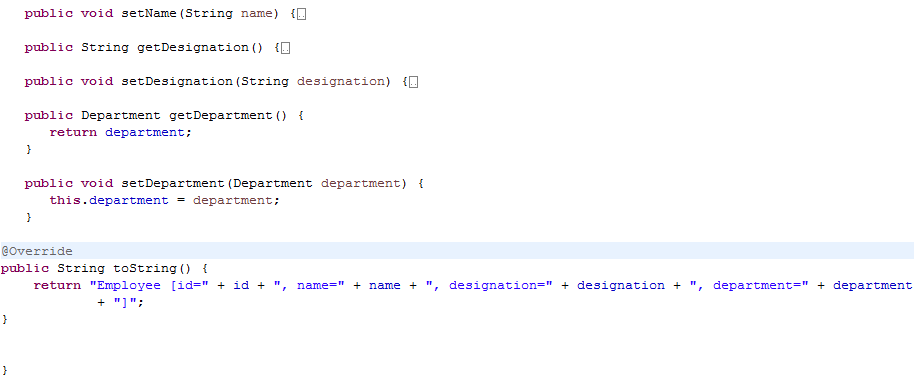
To define a  one-to-many bidirectional association, the @ManyToOne annotation must be used in the target entity (i.e.  Employee entity) and the mappedBy attribute must be used in the source entity (i.e.  Department entity).

Modify the code of Department and Employee classes as follows.

**Department.java**

**** ****

**Employee.java**

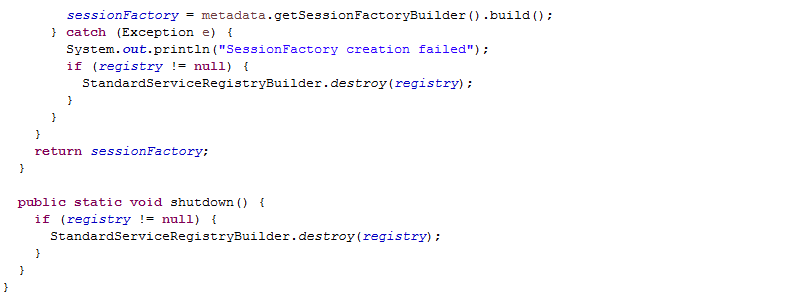
 

**Hibernate utility class**

Create a helper class HibernateUtil to bootstrap hibernate.

Map the Employee and Address entities using the #MetadataSources.addAnnotatedClass() method.

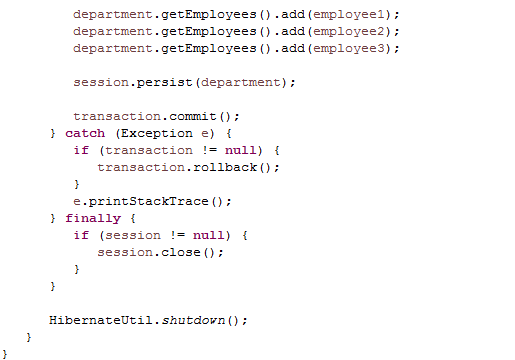
**HibernateUtil.java**

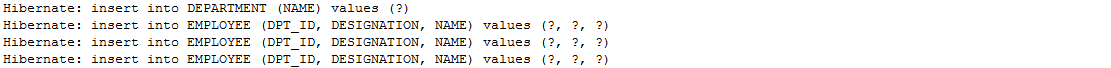
**Main class**

Create the MainApp class to run the application.

**MainApp.java**

**Output**

On executing the MainApp class, you will see the following records in DEPARTMENT and EMPLOYEE tables.

